COURSE SYLLABUS AND COURSE REQUIREMENTS
ACADEMIC YEAR 2023/2024 I. SEMESTER FALL

Course title	*Web Based Software Development*
Course Code	IVB474AN
Hours/Week: le/pr/lab	1/0/2
Credits	4
Degree Programme	Computer Science Engineering BSC
Study Mode	*full time course*
Requirements	mid-term grade
Teaching Period	autumn
Prerequisites	Databases I, Programming 3
Department(s)	System and Software Technology
Course Director	Etelka Szendrői Dr.
Teaching Staff	*Etelka Szendrői Dr.*

COURSE DESCRIPTION

The course serves to further deepen programming knowledge. It provides insight into the world of web application development using ASP.NET Core MVC technology. During the course, ASP.NET Core technology, the structure of ASP.NET Core MVC applications will be presented. Access to the data stored in the database is realized using the capabilities of the Entity Framework Core object-relational mapping, They learn how to use CRUD operation to manipulate data stored in relational database.

SYLLABUS

1. GOALS AND OBJECTIVES

The key goal is to enable students to create a cross-platform web application using ASP.NET Core. Within the framework of the course, students will get acquainted with the basics and techniques of web-based software development methods. They gain the ability to create data-driven applications in a Microsoft ASP.NET Core development environment using Entity Framework Core technology. During the semester, beyond simple tasks, they develop a complex ASP.NET Core trading application.

2. COURSE CONTENT

	TOPICS
LECTURE	1. *The ASP.NET technology. Http protocol.* 2. *Comparison between ASP.NET and ASP.NET Core. ASP.NET Core MVC architecture, project structure.* 3. *Razor syntax. Razor views. Styles. Create responsive pages, applying Bootstrap. Use Nuget Package Manager.* 4. *Input control, validators.* 5. *Navigating between pages. Menus.* 6. *Transfer data from the controller. State management, session variables, cookie use.* 7. *Data access. Entity Framework Core rchitecture. Migration.* 8. *Dependency Injection.* 9. *Manage relationships between entities in Entity Framework Core.* 10. *CRUD operations, search, filter.* 11. *Fluent API. Repository design pattern.* 12. *ASP.NET Core Identity, User Management, Authentication, Security.*

LABORATORY PRACTICE

1. *Create simple ASP.NET Core application. The MVC project architecture. Configuration settings*
2. *Using validators.*
3. *Styles, Razor syntax, Bootstrap.*
4. *Creating a database connection. Manipulate data.*
5. *Entity Framework Core Code First approach. Create model classes. The DBContext class. Migration.*
6. *Populate the database with initial data. Seed method.*
7. *Create the Controller classes for the application.*
8. *Razor Views*
9. *Design navigation. State management.*
10. *EF Core CRUD operations.*
11. *Security, using ASP.NET Core Identity. Creating roles and users.*

DETAILED SYLLABUS AND COURSE SCHEDULE

LECTURE

week	**Topic**	**Compulsory reading; page number (from … to …)**	**Required tasks (assignments, tests, etc.)**	**Completion date, due date**
1.	The ASP.Net and ASP.NET Core technology. HTTP protocol. The architecture of ASP.NET Core MVC applications.	[1] 4-30	…	…
2.	Configuration settings of ASP.NET Core projects. Razor View. HttpGET and HttpPost methods.	[1] 38-75		
3.	Build responsive web pages. Styles, Bootstrap classes. Navigation menu. Control user inputs, valudation.	[1] 84-125		
4.	Creating data-driven ASP.NET Core MVC application. Entity Framework Core technology. Connection string. DBContext class. Migration.	[1] 134-170		
5.	Entity Framework Core CRUD operations. LINQ.	[1] 134-170		
6.	MID-TERM TEST		TEST 1.	
7.	Controller and eouting. Transport data from controller to view. Binding the data model to the View.	[1]204-222		
8.	Session state and cookies usage. Tag helpers. Client-side validation.	[1] 330-356		
9.	Dependency injection, Repository design pattern.	[1] 560-570		
10.	Manage relationships between entities in Entity Framework Core.	[1] 446-466		
11.	ASP.NET Core Identity. Manage users and roles. Authenticate users.	[1] 648-666		
12.	ASP.NET Core Identity. Manage users and roles. Authenticate users.	[1] 648-666		
13.	**FINAL TEST**		TEST2	

PRACTICE, LABORATORY PRACTICE

week	**Topic**	**Compulsory reading; page number (from … to …)**	**Required tasks (assignments, tests, etc.)**	**Completion date, due date**
1.	The development environment. ASP.NET Core project structure.			
2.	Building a simple ASP.NET Core MVC application.			
3.	Control input, validation. Creating model classes.			
4.	Building database connection, manipulate data in Entity Framework Core.			
5.	Css files. Using Bootstrap classes to build responsive web pages. Layout.			
6.	MIDTERM-Test		TEST 1.	10th of October
7.	Creating Bookstore ASP.NET Core MVC project. Configuration settings. Create model classes.			
8.	Applying Entity Framework Core Code First approach. Build the database connection. DBContext class. Migration. Create database, populate initial data with Seed method.		Semester homework	Due date: 1th week of exam period
9.	Controller classes. Create Views and navigation menu. Building View-models.			
10.	Area of administrators. Creating Model, Controller classes and Views for administrative tasks.			
11.	Creating shopping cart. Managing cartoperations.			
12.	ASP.NET Core Identity. Roles, users and authentication.			
13.	FINAL TEST		TEST 2	28th November

3. ASSESSMENT AND EVALUATION

ATTENDANCE

In accordance with the Code of Studies and Examinations of the University of Pécs, Article 45 (2) and Annex 9. (Article 3) a student may be refused a grade or qualification in the given full-time course if the number of class absences exceeds 30% of the contact hours stipulated in the course description.

Method for monitoring attendance *(e.g.: attendance sheet / online test/ register, etc.)*

attendance sheet

ASSESSMENT

Course resulting in mid-term grade *(PTE TVSz 40§(3))*

Mid-term assessments, performance evaluation and their ratio in the final grade *(The samples in the table to be deleted.)*

Type	Assessment	Ratio in the final grade
Midterm-Test	*100%*	*25 %*
Final Test	*100%*	*55 %*
Semester homework	*max 20 points (100%)*	*20 %*

Opportunity and procedure for re-takes (PTE TVSz 47§(4))

The specific regulations for improving grades and resitting tests must be read and applied according to the general Code of Studies and Examinations. E.g.: all tests and assessment tasks can be repeated/improved at least once every semester, and the tests and home assignments can be repeated/improved at least once in the first two weeks of the examination period.

The average of midterm-test and the final test must be equal or higher than 40%. If the average percent is lower than 40% should be retake the test in one time at the first week of exam period.
Presentation of Semester homework will be in the first week of exam period.

Grade calculation as a percentage

based on the aggregate performance according to the following table

Course grade	Performance in %
excellent (5)	85 % …
good (4)	70 % ... 85 %
satisfactory (3)	55 % ... 70 %
pass (2)	40 % ... 55 %
fail (1)	below 40 %

The lower limit given at each grade belongs to that grade.

4. SPECIFIED LITERATURE

In order of relevance. (In Neptun ES: Instruction/Subject/Subject details/Syllabus/Literature)

COMPULSORY READING AND AVAILABILITY

[1.] Mary Delamater, Joel Murach, *Murach's ASP.NET Core MVC*, Mike Murach & Associates, Inc. 2020, ISBN: 978-1-943872-49-7
[2.] Andrew Lock, ASP.NET Core in Action, Second edition, Manning Publication Co., 2021, ISBN: 978-1-61729-830-1

RECOMMENDED LITERATURE AND AVAILABILITY

[3] Adam Freeman, Pro Entity Frmework Core 2 for ASP.NET Core MVC, Apress, 2018, ISBN: 978 1 4842 3435 8
[4] BRIAN L. GORMAN, *PRACTICAL ENTITY FRAMEWORK*, APRESS 2020, ISBN:978 1 4842 6044 9